

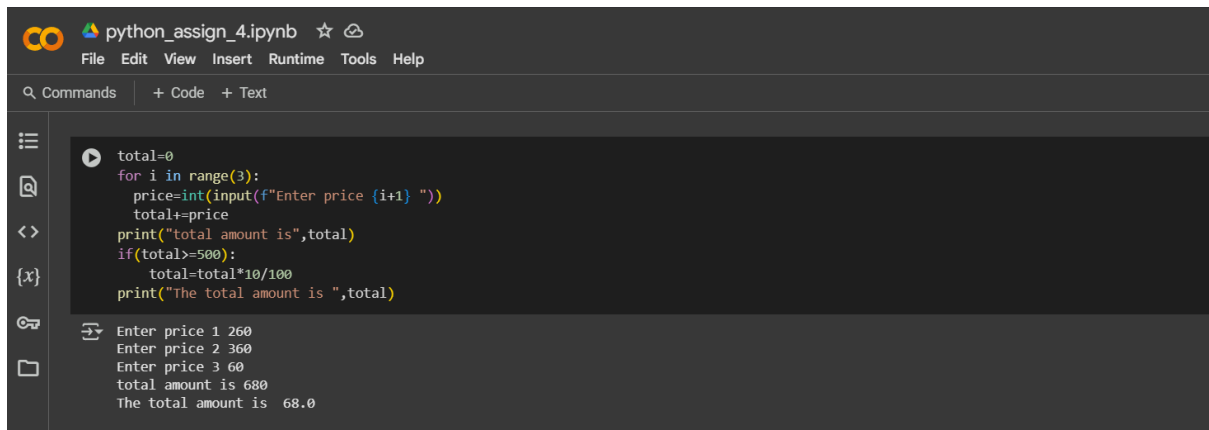
Name: B Bharanidharan

Reg. No: 24MCR010

Class: I – MCA – “A”

Python Assignment - 4

Case Study-1:



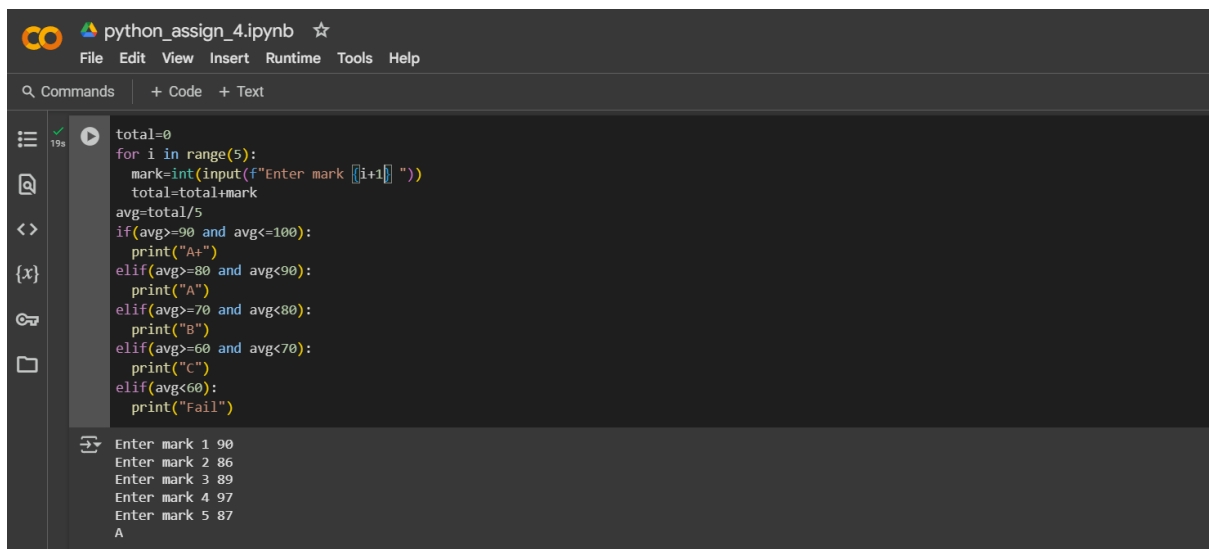
```
python_assign_4.ipynb
File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

total=0
for i in range(3):
    price=int(input(f"Enter price {i+1} "))
    total+=price
print("total amount is",total)
if(total>=500):
    total=total*10/100
print("The total amount is ",total)

Enter price 1 260
Enter price 2 360
Enter price 3 60
total amount is 680
The total amount is  68.0
```

Case Study – 2:



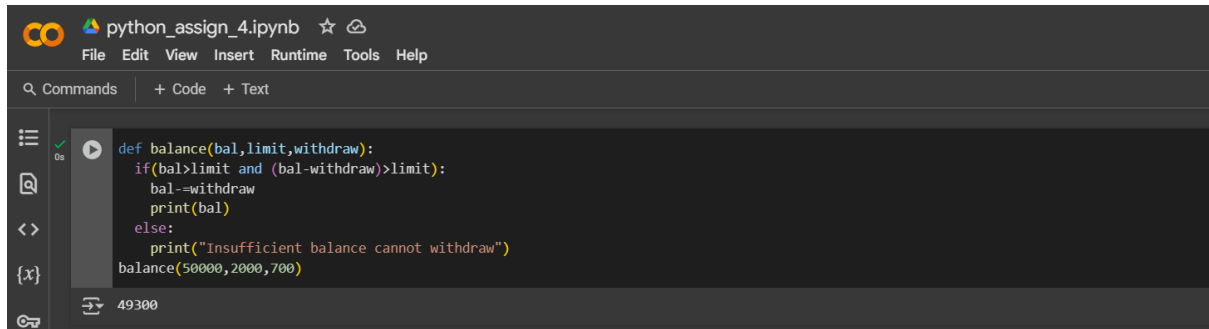
```
python_assign_4.ipynb
File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

total=0
for i in range(5):
    mark=int(input(f"Enter mark {i+1} "))
    total=total+mark
avg=total/5
if(avg>=90 and avg<=100):
    print("A+")
elif(avg>=80 and avg<90):
    print("A")
elif(avg>=70 and avg<80):
    print("B")
elif(avg>=60 and avg<70):
    print("C")
elif(avg<60):
    print("Fail")

Enter mark 1 90
Enter mark 2 86
Enter mark 3 89
Enter mark 4 97
Enter mark 5 87
A
```

Case Study – 3:



The image shows a Jupyter Notebook interface with a dark theme. The title bar at the top reads "python_assign_4.ipynb" and includes icons for file operations, a star, and a share icon. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A search bar labeled "Commands" is on the left, followed by buttons for "+ Code" and "+ Text". The main area displays a Python function definition for a withdrawal process. The function takes three arguments: 'bal' (current balance), 'limit' (maximum withdrawal limit), and 'withdraw' (amount to withdraw). It checks if the balance is greater than the limit and if the remaining balance after withdrawal is still greater than the limit. If both conditions are met, it prints the new balance and updates 'bal'. Otherwise, it prints an error message and calls the function again with initial values (50000, 2000, 700). The function is called with 'balance(50000, 2000, 700)'. The output at the bottom shows the value 49300.

```
def balance(bal,limit,withdraw):  
    if(bal>limit and (bal-withdraw)>limit):  
        bal-=withdraw  
        print(bal)  
    else:  
        print("Insufficient balance cannot withdraw")  
        balance(50000,2000,700)  
  
balance(50000,2000,700)
```

49300

Case Study – 6:

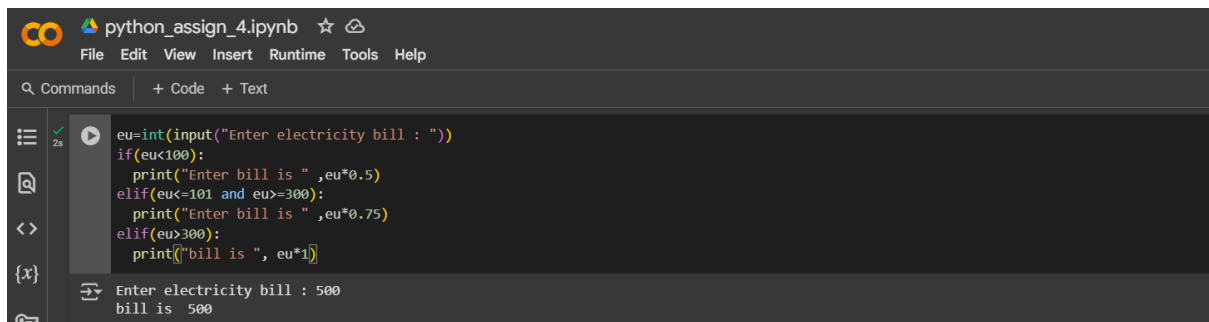


The image shows a Jupyter Notebook interface with a dark theme. The title bar at the top reads "python_assign_4.ipynb" and includes icons for file operations, a star, and a share icon. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A search bar labeled "Commands" is on the left, followed by buttons for "+ Code" and "+ Text". The main area displays a Python program that calculates the amount each friend should pay after a discount. It prompts the user to enter the total bill and the number of friends. If the total bill is greater than 200, a 10% discount is applied. The final amount is divided by the number of friends. The output shows the results for a total bill of 15000 and 6 friends.

```
total_bill=int(input("Enter total bill : "))  
friends=int(input("Enter number of friends : "))  
if(total_bill > 200):  
    total_bill=total_bill*10/100  
    print("The amount after discount is :",total_bill)  
total_bill=total_bill/friends  
print("Each one should pay this amount ",total_bill)
```

Enter total bill : 15000
Enter number of friends : 6
The amount after discount is : 1500.0
Each one should pay this amount 250.0

Case Study – 7:



The image shows a Jupyter Notebook interface with a dark theme. The title bar at the top reads "python_assign_4.ipynb" and includes icons for file operations, a star, and a share icon. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A search bar labeled "Commands" is on the left, followed by buttons for "+ Code" and "+ Text". The main area displays a Python program that calculates the electricity bill based on usage. It prompts the user to enter the electricity bill. If the bill is less than 100, a 5% discount is applied. If the bill is between 101 and 300, a 7.5% discount is applied. If the bill is greater than 300, no discount is applied. The output shows the results for a bill of 500.

```
eu=int(input("Enter electricity bill : "))  
if(eu<100):  
    print("Enter bill is ",eu*0.5)  
elif(eu<=101 and eu<=300):  
    print("Enter bill is ",eu*0.75)  
elif(eu>300):  
    print("bill is ", eu*1)
```

Enter electricity bill : 500
bill is 500